UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/691,355	10/21/2003	Thomas Earl Palmer	08223/0200098-US0	6645

7278 7590 09/20/2007
DARBY & DARBY P.C.
P.O. BOX 770
Church Street Station
New York, NY 10008-0770

EXAMINER	
HA, LEYNNA A	
ART UNIT	**PAPER NUMBER**
2135	

MAIL DATE	DELIVERY MODE
09/20/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No. 10/691,355	Applicant(s) PALMER, THOMAS EARL	
	Examiner LEYNNA T. HA	Art Unit 2135	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on 29 June 2007.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) 1-47 is/are pending in the application.

4a) Of the above claim(s) ____ is/are withdrawn from consideration.

5)☐ Claim(s) ____ is/are allowed.

6)☒ Claim(s) 1-47 is/are rejected.

7)☐ Claim(s) ____ is/are objected to.

8)☐ Claim(s) ____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on ____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. ____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☒ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date 9/13/07.

4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. ____.

5) ☐ Notice of Informal Patent Application

6) ☐ Other: ____.

DETAILED ACTION

1. Claims 1-47 are pending.

2. Claims 1-47 previously rejected under 35 U.S.C. 101, is now withdrawn.

Response to Arguments

3. Applicant's arguments with respect to claims 1-47 have been considered but are moot in view of the new ground(s) of rejection.

Examiner traverses the argument on pg.12, that McDonough does not suggest selecting a direction based on a generated bit sequence. McDonough discloses generating one or more data sequences for communications includes steps of providing data at an input of a memory which stores bits associated with a PN sequence (bit sequence), changing the data, and for each of a plurality of changes of the data, providing the selected PN bit of the PN sequence at an output of said memory based on the data (col.6, line 65 – col.7, line 4). McDonough also discusses data of a stored data sequence that corresponds to the memory location data is selected and the selected data is provide for use in communication where the stored data sequence is a PN sequence (col.10, lines 41-67). Hence, McDonough obviously reads on the claimed selecting a direction based on a generated bit sequence.

The argument on p.13, is traversed where McDonough does not suggest determining an offset between a cursor position and a match bit within the n-dimensional entity. Specification (pg.11), vaguely explains the cursor position may be converted into a count of the number of pseudo-random numbers that are employed to

substantially the same view under the cursor position and specification further discusses a cursor normalizer. Based on the specification and the claimed, a cursor position is broad where the cursor position is not specific or limited to what constitute the claimed cursor position. Thus, the counter data or counter value of McDonough (col.9, line 58 – col.10, line 3) reads closest based on the specification.

As per dependent claims, they are rejected by virtue of their dependency.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over McDonough (US 6,549,563), and further in view of Zdepski, et al. (US 6,606,746).

As per claim 1:

McDonough method of encrypting a data string, comprising:

<u>receiving, from the user, the data string, as plaintext;</u> **(McDonough on col.13, lines 64-67 and ZDepski on col.6, lines 49-62)**

generating an n-dimensional entity, wherein the n-dimensional entity comprises random bits; and **(col.6, lines 65 – col.7, line 1; the n-dimensional entity can**

broadly be given in light as a data sequence as disclosed by McDonough and the claimed bit sequence refers to a pseudorandom noise (PN) sequence (col.8, lines 17-19 and col.9, lines 48-55) where the N-bit counter for generating a PN sequence of length 2N.)

for each bit in the data string:

reading a number of bits from the n-dimensional entity; **(col.4, lines 45-46)**

performing an action based in part on the read number of bits; **(col.9, lines 33-39 and col.11, lines 21-27; McDonough discusses data sequence that was said above to have N-bits to generate the PN sequence where each data sequence may be quite different from one another in terms of length and/or purpose. Hence, the purpose based on the bits is the claimed an action.)**

generating a bit sequence; **(col.8, lines 17-19)**

selecting a direction within the n-dimensional entity based in part on the generated bit sequence; **(col.10, lines 45-63 and col.11, lines 5-25)**

determining an offset between a cursor position and a match bit within the n-dimensional entity, wherein the match bit is based in part on the action, the direction, and the each bit in the data string; an **(col.11, lines 5- 35 and col.12, lines 23-67)**

modifying the generated bit sequence **(col.7, lines 3-4)** with the determined offset **(col.10, lines 43-50 and col.11, lines 65-66)** *<u>[by inserting the determined offset into the generated bit sequence</u>* **(ZDepski on col.17, 62-65)***<u>]</u>* to generate <u>a row</u> **(col.3,**

lines 49-51 and col.17, lines 15-46) *[within an encoded data string]* *corresponding to each bit in the data string;*

storing the resulting *[encoded]* data string as an encoded representation of the received plaintext data string. **(col.10, lines 65-67 and col.11, lines 65-67)**

McDonough discusses changing the generated bit sequence with the determined offset and encoding the signals appropriately but did not provide further the limitation by inserting the determined offset into bit sequence within an encoded data string.

Zdepski discloses the method provides an insert picture to the encoder where the method encodes the insert picture to create the compressed insert picture. The encoder preferably uses the background picture slice encoding information in encoding the insert picture in order to place the correct horizontal and vertical position data at the beginning of each slice in the compressed insert picture. Zdepski discusses the method optionally creates an insert picture slice map indicating locations for the one or more slices comprising the compressed insert picture in the compressed background picture. Further, in the preferred embodiment, the insert picture slice map comprises one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (col.17, lines 35-65). Zdepski discloses a software encoder, which accepts the background picture slice encoding information, is used to encode the background picture. The insertion video sequence is also compressed, using the software encoder, with a row of macroblocks in each slice. Slice maps are then created for each of the compressed

background picture and the compressed insert picture. Creation of the slice maps comprises determining the byte offsets into the bitstreams to point to the slice start codes in each picture. A mutliplexed signal is then created comprising: a) an interactive application to perform the pasting operations; b) the background and insert picture bitstreams; and c) the slice maps or byte offsets into the background and insert bitstreams. This multiplexed stream can then be transmitted to one or more subscriber televisions (col.19, line 57 – col.20, line 4).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of McDonough with Zdepski teaching inserting the determined offset into bit sequence within an encoded data string because one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (Zdepski - col.17, lines 35-65) where this multiplexed stream can then be transmitted to one or more subscriber televisions (Zdepski - col.19, line 57 – col.20, line 4).

As per claim 2: the method of claim 1, wherein generating the n-dimensional entity further comprises: generating a seed for a random number generator **(Rhoads on col.2, lines 63-65)**; determining a number of dimensions of the n-dimensional entity; determining a length for each dimension of the n-dimensional entity **(McDonough on col.3, lines 50-55)**; and populating the n-dimensional entity with bits from the random number generator. **(McDonough on col.4, lines 30-50 and col.9, lines 48-55)**

As per claim 3: see McDonough on col.1, lines 58-59; discussing the method of claim 2, wherein the number of dimensions is determined based in part on at least one of a user selectable input, a default value, and a random number.

As per claim 4: see McDonough on col.1, lines 58-59; discussing the method of claim 2, wherein the length of each dimensions is determined based in part on at least one of a user selectable input, a default value, and a random number.

As per claim 5: see McDonough on col.1, lines 58-59 and col.7, line 1; discussing the method of claim 2, wherein the random number generator is arranged to produce a pseudo-random bit sequence.

As per claim 6: see McDonough on col.4, lines 45-46 and col.10, lines 52-53; discussing the method of claim 1, wherein reading the number of bits from the n-dimensional entity further comprises reading a sequence of bits equal to a size of an op-code.

As per claim 7: see McDonough on col.10, lines 43-54 and ZDepski on col.6, lines 49-62; discussing the method of claim 6, wherein the size of the op-code is selected from at least one of a default size and a user selectable input.

As per claim 8: see McDonough on col.4, lines 45-46 and col.10, lines 52-53; discussing the method of claim 1, wherein performing the action further comprises a means for associating an action to the read number of bits.

As per claim 9 see McDonough on col.9, lines 42-56 and col.10, lines 52-53; discussing the method of claim 1, wherein performing the action further comprises:

interpreting the read number of bits as an op-code; determining an action associated with the op-code; and executing the action associated with the op-code.

As per claim : see McDonough on col.4, lines 31 and 45 and col.10, lines 52-53; discussing the method of claim 1, wherein performing the action further comprises associating the read number of bits with an action using at least one of a database, a table, a linked-list, and a program.

As per claim 11: see McDonough on col.4, lines 45-46 and col.10, lines 52-53; discussing the method of claim 1, wherein performing the action further comprises performing at least one of changing a cursor position, switching a bit state, reading a bit, modifying a bit, generating another n-dimensional entity, changing a direction, and modifying an interpretation of a bit state.

As per claim 12: see McDonough on col.1, lines 58-59 and col.7, line 1; discussing the method of claim 1, wherein generating a bit sequence further comprises generating a truly random bit sequence.

As per claim 13: see McDonough on col.10, lines 43-54 and ZDepski on col.6, lines 49-62; discussing the method of claim 1, further comprising, combining an encoded data string associated with a bit in the data string with another encoded data string associated with a different bit in the data string.

As per claim 14: see McDonough on col.1, lines 63-65 and ZDepski on col.6, lines 49-62; discussing the method of claim 1, further comprising exclusive or-ing each encoded data string with a previous encoded data string, wherein a first encoded data string is exclusively or-ed with a last encoded data string.

As per claim 15: see McDonough on col.10, lines 43-54 and ZDepski on col.6, lines 49-62; discussing the method of claim 1, further comprising combining bits within an encoded data string with a corresponding bit within an obfuscation table.

As per claim 16: see McDonough on col.1, lines 58-67 and ZDepski on col.6, lines 49-62; discussing the method of claim 1, further comprising modifying the length of at least one encoded data string.

As per claim 17: see McDonough on col.7, line 1 and col.11, lines 65-66; discussing the method of claim 1, further comprising including at least one random data string with the each encoded data string.

As per claim 18: see McDonough on col.12, line 23-50; discussing the method of claim 1, wherein generating the n-dimensional entity further comprises determining a fingerprint associated with a computing system in which the method operates.

As per claim 19:

A method of encrypting a data string, comprising:

<u>receiving, from the user, the data string to be encoded;</u> **(McDonough on col.11, lines 65-67 and col.13, lines 64-67 and ZDepski on col.6, lines 49-62)**

generating an n-dimensional entity, wherein the n-dimensional entity is populated with pseudo-random bits; **(col.6, lines 65 – col.7, line 1; the n-dimensional entity can broadly be given in light as a data sequence as disclosed by McDonough and the claimed bit sequence refers to a pseudorandom noise (PN) sequence (col.8, lines 17-19 and col.9, lines 48-55) where the N-bit counter for generating a PN sequence of length 2N.)**

for each bit in the data string:

determining a cursor position within the n-dimensional entity; determining a direction within the n-dimensional entity; **(col.3, lines 49-55 and col.4, lines 42-45)**

determining a number of bits in the n-dimensional entity, wherein the bits are read from the determined cursor position along the determined direction; **(col.7, lines 3-4)**

performing an action based in part on the determined number of bits; **(col.7, lines 3-4)**

performing an action based in part on the read number of bits; **(col.9, lines 33-39 and col.11, lines 21-27; McDonough discusses data sequence that was said above to have N-bits to generate the PN sequence where each data sequence may be quite different from one another in terms of length and/or purpose. Hence, the purpose based on the bits is the claimed an action.)**

generating a bit sequence; **(col.8, lines 17-19)**

selecting another direction based in part on the bit sequence; **(col.9, lines 4-12 and col.11, lines 5-25)**

determining an offset between a match bit within the n-dimensional entity and the cursor position, wherein the match bit is based in part on the action, the other direction, and the each bit in the data string; and

modifying the bit sequence **(col.7, lines 3-4)** with the determined offset **(col.10, lines 43-50 and col.11, lines 65-66)** *[by inserting the determined offset into the*

<u>*generated bit sequence*</u> *to generate an encoded data string for each bit in the data string* ***(ZDepski on col.17, 62-65)];*** and

<u>storing the resulting *[encoded]* data string as an encoded representation of the received plaintext data string.</u> **(col.10, lines 65-67 and col.11, lines 65-67)**

McDonough discusses changing the generated bit sequence with the determined offset and encoding the signals appropriately but did not provide further the limitation by inserting the determined offset into the generated bit sequence to generate an encoded data string for each bit in the data string.

Zdepski discloses the method provides an insert picture to the encoder where the method encodes the insert picture to create the compressed insert picture. The encoder preferably uses the background picture slice encoding information in encoding the insert picture in order to place the correct horizontal and vertical position data at the beginning of each slice in the compressed insert picture. Zdepski discusses the method optionally creates an insert picture slice map indicating locations for the one or more slices comprising the compressed insert picture in the compressed background picture. Further, in the preferred embodiment, the insert picture slice map comprises one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (col.17, lines 35-65). Zdepski discloses a software encoder, which accepts the background picture slice encoding information, is used to encode the background picture. The insertion video sequence is also compressed, using the software encoder, with a row of macroblocks in each slice. Slice maps are then created for each of the compressed

background picture and the compressed insert picture. Creation of the slice maps comprises determining the byte offsets into the bitstreams to point to the slice start codes in each picture. A mutliplexed signal is then created comprising: a) an interactive application to perform the pasting operations; b) the background and insert picture bitstreams; and c) the slice maps or byte offsets into the background and insert bitstreams. This multiplexed stream can then be transmitted to one or more subscriber televisions (col.19, line 57 – col.20, line 4).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of McDonough with Zdepski teaching by inserting the determined offset into the generated bit sequence string because one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (Zdepski - col.17, lines 35-65) where this multiplexed stream can then be transmitted to one or more subscriber televisions (Zdepski - col.19, line 57 – col.20, line 4).

As per claim 20: see McDonough on col.2, lines 59-64 and col.6, line 65-col.7, line 1 ; discussing the method of claim 19, wherein generating the bit sequence further comprises generating a truly random bit sequence.

As per claim 21: see McDonough on col.4, lines 45-46; discussing the method of claim 19, wherein determining the number of bits in the n-dimensional entity, further comprises a means for determining an action based in part on the read number of bits.

As per claim 22: see McDonough on col.10, lines 50-55; discussing the method of claim 19, wherein performing an action further interpreting the determined number of bits as an op-code; and executing an action associated with the op-code.

As per claim 23: see McDonough on col.10, lines 43-54 and ZDepski on col.6, lines 49-62; discussing the method of claim 19, further comprising employing an obfuscation table to obfuscate the encoded data string for each bit in the data string.

As per claim 24: see McDonough on col.8, lines 10-54; discussing the method of claim 19, wherein determining the cursor position further receiving a cursor position; and normalizing the received cursor position to within a boundary of the n-dimensional entity.

As per claim 25: see McDonough on col.7, lines 33-45; discussing the method of claim 24, wherein normalizing the received cursor position further comprises employing a circular orbiting algorithm to the cursor position until the cursor position is within the boundary of the n-dimensional entity.

As per claim 26: see McDonough on col.8, lines 3-7; discussing the method of claim 19, wherein selecting another direction further comprises employing a predetermined set of bits in the bit sequence to select the other direction.

As per claim 27: see McDonough on col.10, lines 43-50; discussing the method of claim 19, wherein modifying the bit sequence with the offset further comprises overwriting a predetermined set of bits in the bit sequence with the determined offset.

As per claim 28: see McDonough on col.4, lines 1-50; discussing the method of claim 19, wherein determining the offset further comprises generating another n-dimensional entity, if the match bit is not located.

As per claim 29: see McDonough on col.3, lines 33-34 and col.4, lines 1-50; discussing the method of claim 19, wherein determining the offset further comprises setting a bit in the bit sequence, if the match bit is not located.

As per claim 30: see McDonough on col.1, lines 56-67 and ZDepski on col.6, lines 49-62; discussing the method of claim 19, wherein generating the n-dimensional entity, further comprises: generating a fingerprint based in part on a computing system in which the method operates; and determining a characteristic of the n-dimensional entity based in part on the fingerprint.

As per claim 31: see McDonough on col.2, lines 59-64 and col.3, lines 30-34; discussing the method of claim 30, wherein the characteristic of n-dimensional entity further comprises at least one of a length of a side, a number of dimensions, and a seed for a random number generator which is enabled to populate the n-dimensional entity with random bits.

As per claim 32: see McDonough on col.5, lines 22-25; discussing the method of claim 30, wherein the fingerprint further comprises a hash of at least one of a Central Processing Unit's (CPU's) kernel speed, CPU serial number, CPU family identity, CPU manufacturer, an operating system globally unique identifier (GUID), a hardware component enumeration, Internet Protocol (IP) address, BIOS serial number, disk serial number, kernel version number, operating system version number, operating system

build number, machine name, installed memory characteristic, physical port enumeration, customer supplied ID, and a MAC address.

As per claim 33: see McDonough on col.9, lines 58-65; discussing the method of claim 32, wherein the hash further comprises at least one a Message Digests (MD), a secure hash, and a Secure Hash Algorithm (SHA).

As per claim 34: see McDonough on col.2, lines 59-64 and col.9, lines 58-65; discussing the method of claim 19, wherein generating the n-dimensional entity, further comprises: creating a digest in part from a fingerprint associated with a computing system in which the method operates; seeding a pseudo-random number generator in part with the digest; determining a number of dimensions of the n-dimensional entity based in part on an output of the pseudo-random number generator; and determining a length of a side of the n-dimensional entity based in part on another output of the pseudo-random number generator.

As per claim 35: see McDonough on col.2, lines 59-64 and col.9, lines 58-65; discussing the method of claim 34, wherein creating the digest further comprises, combining the fingerprint with a user seed to create the digest.

As per claim 36:

A system for encrypting a data string, comprising:

an entity generator that is arranged to generate an n-dimensional entity; **(col.6, lines 65 – col.7, line 1; the n-dimensional entity can broadly be given in light as a data sequence as disclosed by McDonough and the claimed bit sequence refers**

to a pseudorandom noise (PN) sequence (col.8, lines 17-19 and col.9, lines 48-55) where the N-bit counter for generating a PN sequence of length 2N.)

a mapper, arranged to receive the n-dimensional entity, and perform actions, comprising:

bits; **(col.8, lines 17-19)**

receiving a data string; and **(col.9, lines 46-54)**

for each bit in the data string:

reading a number of bits from the n-dimensional entity; **(col.4, lines 45-46)**

performing an action based in part on the read number of generating a bit sequence; **(col.9, lines 33-39 and col.11, lines 21-27; McDonough discusses data sequence that was said above to have N-bits to generate the PN sequence where each data sequence may be quite different from one another in terms of length and/or purpose. Hence, the purpose based on the bits is the claimed an action.)**

selecting a direction within the n-dimensional entity based in part on the generated bit sequence; **(col.9, lines 4-12 and col.11, lines 5-25)**

determining an offset between a cursor position and a match bit within the n-dimensional entity, wherein the match bit is based in part on the action, the direction, and the each bit in the data string; and **(col.10, lines 11-21 and col.11, lines 33-38)**

modifying the generated bit sequence **(col.7, lines 3-4)** *[by inserting the generated bit sequence into the determined offset to generate an encode data string* ***(ZDepski on col.17, 62-65)****]*; and **(col.10, lines 43-50 and col.11, lines 65-66)**

storing the *[encoded]* data string in a data store as an encoded representation of the received plaintext data string. **(col.10, lines 65-67 and col.11, lines 65-67)**

McDonough discusses changing the generated bit sequence with the determined offset and encoding the signals appropriately but did not provide further the limitation by inserting the generated bit sequence into the determined offset to generate an encode data string.

Zdepski discloses the method provides an insert picture to the encoder where the method encodes the insert picture to create the compressed insert picture. The encoder preferably uses the background picture slice encoding information in encoding the insert picture in order to place the correct horizontal and vertical position data at the beginning of each slice in the compressed insert picture. Zdepski discusses the method optionally creates an insert picture slice map indicating locations for the one or more slices comprising the compressed insert picture in the compressed background picture. Further, in the preferred embodiment, the insert picture slice map comprises one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (col.17, lines 35-65). Zdepski discloses a software encoder, which accepts the background picture slice encoding information, is used to encode the background picture. The insertion video sequence is also compressed, using the software encoder, with a row of macroblocks in each slice. Slice maps are then created for each of the compressed background picture and the compressed insert picture. Creation of the slice maps comprises determining the byte offsets into the bitstreams to point to the slice start

codes in each picture. A mutliplexed signal is then created comprising: a) an interactive application to perform the pasting operations; b) the background and insert picture bitstreams; and c) the slice maps or byte offsets into the background and insert bitstreams. This multiplexed stream can then be transmitted to one or more subscriber televisions (col.19, line 57 – col.20, line 4).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of McDonough with Zdepski teaching by inserting the generated bit sequence into the determined offset because one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (Zdepski - col.17, lines 35-65) where this multiplexed stream can then be transmitted to one or more subscriber televisions (Zdepski - col.19, line 57 – col.20, line 4).

As per claim 37: the system of claim 36, wherein the entity generator generates the n-dimensional entity by performing actions, comprising: determining a seed for a random number generator **(Rhoads on col.2, lines 63-65)**; determining a number of dimensions of the n-dimensional entity; determining a length for each dimension of the n-dimensional entity **(McDonough on col.3, lines 50-55)**; and populating the n-dimensional entity with bits from the random number generator. **(McDonough on col.4, lines 30-50 and col.9, lines 48-55)**

As per claim 38: see McDonough on col.10, lines 41-43; discussing the system of claim 37, wherein determining the seed further comprises creating the seed from a

combination of a user seed and a fingerprint that is associated with a computing system in which the system operates.

As per claim 39: see McDonough on col.4, lines 1-50 and col.7, lines 9-18; discussing the system of claim 37, wherein the initial cursor position is determined based in part on normalizing a received cursor position to within a boundary of the n-dimensional entity.

As per claim 40: see McDonough on col.2, lines 59-64; discussing the system of claim 37, wherein the number of dimensions is determined based in part on at least one of a user selectable input, a default value, and a random number.

As per claim 41: see McDonough on col.7, lines 9-55; discussing the system of claim 36, wherein the generated n-dimensional entity is populated with pseudo-random bits.

As per claim 42: see McDonough on col.4, lines 45-46 and col.10, lines 42-50; discussing the system of claim 36, wherein performing the action further comprises performing at least one of changing a cursor position, switching a bit state, reading a bit, generating another n-dimensional entity, changing a direction, and modifying an interpretation of a bit state.

As per claim 43: see McDonough on col.4, lines 39-40 and col.10, lines 42-50; discussing the system of claim 36, wherein generating a bit sequence further comprises generating a truly random bit sequence.

As per claim 44:

McDonough discloses an apparatus for encrypting a data string, comprising:

a transceiver that receives the data string and sends an encoded array; **(McDonough on col.11, lines 65-67 and col.13, lines 64-67 and ZDepski on col.6, lines 49-62)**

coupled to the transceiver, an n-dimensional encrypter that is arranged to perform actions, comprising:

generating an n-dimensional entity, wherein the n-dimensional entity comprises random bits; and**(col.6, lines 65 – col.7, line 1; the n-dimensional entity can broadly be given in light as a data sequence as disclosed by McDonough and the claimed bit sequence refers to a pseudorandom noise (PN) sequence (col.8, lines 17-19 and col.9, lines 48-55) where the N-bit counter for generating a PN sequence of length 2N.)**

for each bit in the received data string:

reading a number of bits from the n-dimensional entity; **(col.4, lines 45-46)**

performing an action associated with the read number of bits; **(col.9, lines 33-39 and col.11, lines 21-27; McDonough discusses data sequence that was said above to have N-bits to generate the PN sequence where each data sequence may be quite different from one another in terms of length and/or purpose. Hence, the purpose based on the bits is the claimed an action.)**

generating a bit sequence; **(col.8, lines 17-19)**

selecting a direction within the n-dimensional entity based in part on the generated bit sequence; **(col.10, lines 11-21 and col.11, lines 33-38)**

determining an offset between a cursor position and a match bit within the n-dimensional entity, wherein the match bit is based in part on the action, the direction, and the each bit in the received data string; and **(col.9, lines 4-12 and col.11, lines 5-25)**

modifying the generated bit sequence **(col.7, lines 3-4)** with the determined offset *[by inserting the determined offset into the generated bit sequence to generate an encoded data string, wherein the encoded* ***(ZDepski on col.17, 62-65)]*** data string represents a row within the *[encoded]* array. **(col.10, lines 43-50 and col.11, lines 65-66)**

McDonough discusses changing the generated bit sequence with the determined offset and encoding the signals appropriately but did not provide further the limitation by inserting the determined offset into the generated bit sequence to generate encoded data string.

Zdepski discloses the method provides an insert picture to the encoder where the method encodes the insert picture to create the compressed insert picture. The encoder preferably uses the background picture slice encoding information in encoding the insert picture in order to place the correct horizontal and vertical position data at the beginning of each slice in the compressed insert picture. Zdepski discusses the method optionally creates an insert picture slice map indicating locations for the one or more slices comprising the compressed insert picture in the compressed background picture. Further, in the preferred embodiment, the insert picture slice map comprises one or more byte offsets indicating locations of each of the one or more slices,

preferably pointers to the slice start codes, in the compressed insert picture (col.17, lines 35-65). Zdepski discloses a software encoder, which accepts the background picture slice encoding information, is used to encode the background picture. The insertion video sequence is also compressed, using the software encoder, with a row of macroblocks in each slice. Slice maps are then created for each of the compressed background picture and the compressed insert picture. Creation of the slice maps comprises determining the byte offsets into the bitstreams to point to the slice start codes in each picture. A mutliplexed signal is then created comprising: a) an interactive application to perform the pasting operations; b) the background and insert picture bitstreams; and c) the slice maps or byte offsets into the background and insert bitstreams. This multiplexed stream can then be transmitted to one or more subscriber televisions (col.19, line 57 – col.20, line 4).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of McDonough with Zdepski teaching inserting the determined offset into the generated bit sequence because one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (Zdepski - col.17, lines 35-65) where this multiplexed stream can then be transmitted to one or more subscriber televisions (Zdepski - col.19, line 57 – col.20, line 4).

As per claim 45: see McDonough on col.4, lines 45-46 and col.10, lines 52-53; discussing the apparatus of claim 44, wherein reading the number of bits from the n-

dimensional entity further comprises reading a sequence of bits equal to a size of an op-code.

As per claim 46: see McDonough on col.4, lines 45-46 and col.10, lines 42-50; discussing the apparatus of claim 44, wherein performing the action further comprises performing at least one of changing a cursor position, switching a bit state, reading a bit, modifying a bit, generating another n-dimensional entity, changing a direction, and modifying an interpretation of a bit state.

As per claim 47:

An apparatus of encrypting a data string, comprising:

a means for generating an n-dimensional entity; **(col.6, lines 65 – col.7, line 1; the n-dimensional entity can broadly be given in light as a data sequence as disclosed by McDonough and the claimed bit sequence refers to a pseudorandom noise (PN) sequence (col.8, lines 17-19 and col.9, lines 48-55) where the N-bit counter for generating a PN sequence of length 2N.)**

a means for receiving the data string as input from a user; **(McDonough on col.11, lines 65-67 and col.13, lines 64-67 and ZDepski on col.6, lines 49-62)**

a means for performing an action for each bit in the data string based in part on the n-dimensional entity; **(col.9, lines 33-39 and col.11, lines 21-27; McDonough discusses data sequence that was said above to have N-bits to generate the PN sequence where each data sequence may be quite different from one another in terms of length and/or purpose. Hence, the purpose based on the bits is the claimed an action.)**

a means for generating a random bit sequence **(col.8, lines 17-19)** associated with each bit in the data string; and**(col.9, lines 4-12 and 48-55 and col.11, lines 8-10)**

a means for modifying the each random bit sequence *[by inserting into each random bit sequence* ***(ZDepski on col.17, 62-65)****]* an offset associated with each bit in the data string, wherein the offset is based in part on the action, the n-dimensional entity, and the each bit in the data string; and **(col.10, lines 43-50 and col.11, lines 65-66)**

means for storing the modified bit sequence in a storage device. **(col.10, lines 65-67 and col.11, lines 65-67)**

McDonough discusses changing the generated bit sequence with the determined offset but did not provide further the limitation inserting into each random bit sequence an offset.

Zdepski discloses the method provides an insert picture to the encoder where the method encodes the insert picture to create the compressed insert picture. The encoder preferably uses the background picture slice encoding information in encoding the insert picture in order to place the correct horizontal and vertical position data at the beginning of each slice in the compressed insert picture. Zdepski discusses the method optionally creates an insert picture slice map indicating locations for the one or more slices comprising the compressed insert picture in the compressed background picture. Further, in the preferred embodiment, the insert picture slice map comprises one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (col.17,

lines 35-65). Zdepski discloses a software encoder, which accepts the background picture slice encoding information, is used to encode the background picture. The insertion video sequence is also compressed, using the software encoder, with a row of macroblocks in each slice. Slice maps are then created for each of the compressed background picture and the compressed insert picture. Creation of the slice maps comprises determining the byte offsets into the bitstreams to point to the slice start codes in each picture. A mutliplexed signal is then created comprising: a) an interactive application to perform the pasting operations; b) the background and insert picture bitstreams; and c) the slice maps or byte offsets into the background and insert bitstreams. This multiplexed stream can then be transmitted to one or more subscriber televisions (col.19, line 57 – col.20, line 4).

Therefore, it would have been obvious for a person of ordinary skills in the art to combine the teaching of McDonough with Zdepski teaching inserting into each random bit sequence an offset because one or more byte offsets indicating locations of each of the one or more slices, preferably pointers to the slice start codes, in the compressed insert picture (Zdepski - col.17, lines 35-65) where this multiplexed stream can then be transmitted to one or more subscriber televisions (Zdepski - col.19, line 57 – col.20, line 4).

Conclusion

5. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL.** See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to LEYNNA T. HA whose telephone number is (571) 272-3851. The examiner can normally be reached on Monday - Thursday (7:00 - 5:00PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kim Vu can be reached on (571) 272-3859. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

LHa

KIM VU
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100